3/13 3199022253 #3 DW M01-01

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年 ３月１３日

出 願 番 号
Application Number: 特願２０００－０７３９２６

出 願 人
Applicant (s): 株式会社日立製作所

２０００年１０月 ６日

特 許 庁 長 官
Commissioner,
Patent Office 及 川 耕 造

出証番号 出証特２０００－３０８１６６８

【書類名】　特許願

【整理番号】　H99022531A

【提出日】　平成12年 3月13日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 15/78

【発明者】

【住所又は居所】　東京都国分寺市東恋ケ窪一丁目２８０番地
株式会社日立製作所中央研究所内

【氏名】　荒川　文男

【発明者】

【住所又は居所】　東京都国分寺市東恋ケ窪一丁目２８０番地
株式会社日立製作所中央研究所内

【氏名】　山田　哲也

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社日立製作所

【代理人】

【識別番号】　100075096

【弁理士】

【氏名又は名称】　作田　康夫

【電話番号】　03-3212-1111

【手数料の表示】

【予納台帳番号】　013088

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　データプロセッサ及びデータ処理システム

【特許請求の範囲】

【請求項１】

ＳＩＭＤ型演算器を有するデータプロセッサであって、

前記データプロセッサは、前記ＳＩＭＤ型演算器にベクトルデータの処理を行わせるための命令を有していることを特徴とするデータプロセッサ。

【請求項２】

請求項１において、

前記ＳＩＭＤ型演算器は、浮動小数点数の積和演算を行う複数の演算器を有していることを特徴とするデータプロセッサ。

【請求項３】

命令セット内の命令を実行し、ＳＩＭＤ型演算器を有するデータプロセッサであって、

前記命令セットには、前記ＳＩＭＤ型演算器にベクトルデータの演算を行わせる命令が含まれていることを特徴とするデータプロセッサ。

【請求項４】

請求項３において、

前記ＳＩＭＤ型演算器は、浮動小数点数の積和演算を行う複数の演算器を有していることを特徴とするデータプロセッサ。

【請求項５】

命令セット内の命令を実行するデータプロセッサであって、

前記命令セットには、前記データプロセッサにベクトルデータの内積とスカラーデータとの和を演算させるための命令が含まれていることを特徴とするデータプロセッサ。

【請求項６】

請求項５において、

前記データプロセッサは、４元ベクトルと４元ベクトルとの内積を求め、前記内積とスカラーデータとの和を求める浮動小数点演算器を有していることを特徴

とするデータプロセッサ。

【請求項７】

請求項６において、

前記浮動小数点演算器は、９入力加算器を有していることを特徴とするデータプロセッサ。

【請求項８】

命令セット内の命令を実行するデータプロセッサであって、

前記命令セットには、前記データプロセッサに、行列データとベクトルデータとの積を演算させるための命令が含まれていることを特徴とするデータプロセッサ。

【請求項９】

請求項８において、

前記データプロセッサは、ベクトルデータとベクトルデータとの内積を求める浮動小数点演算器を複数個有していることを特徴とするデータプロセッサ。

【請求項１０】

請求項８において、

前記行列データは４×４の行列データであり、前記ベクトルデータは４元ベクトルであることを特徴とするデータプロセッサ。

【請求項１１】

請求項９において、

前記複数個の浮動小数点演算器のそれぞれは、前記内積とスカラーデータとの和とを求めることが可能な演算器であることを特徴とするデータプロセッサ。

【請求項１２】

請求項１１において、

前記演算器は、９入力加算器を有していることを特徴とするデータ処理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はマイクロプロセッサ等のデータ処理を行うデータプロセッサ、更には

３次元グラフィック制御に好適なデータ処理システムに適用して、効率良く演算並列度を向上させる技術に関するものである。

【０００２】

【従来の技術】

３次元グラフィックス処理を行うデータプロセッサとして、「MICROPROCESSOR REPORT,vol.13,no.5, April.19,1999, pp.1,6-11」では、１命令で４つの浮動小数点積和演算を実行するＳＩＭＤ（Single Instruction Multiple Data）型浮動小数点ユニットを２つ内蔵するプロセッサが示されている。ここで示されているプロセッサは、１つの浮動小数点積和演算で乗算と加算の２演算、４つで８演算、２ユニットでは計１６演算を行うことが可能となっている。更に、上記の２ユニットの他に、通常の浮動小数点積和演算器が２つあるため、更に４演算を行うことが可能であり、合計２０演算を１サイクルで行うことが可能となっている。

【０００３】

また、上記以外のデータプロセッサとして、「IEEE Micro,vol.18,no.2, March/April 1998, pp.26-34」においては、浮動小数点の内積を求めるための命令を有し、１命令で４つの乗算と３つの加算を実行して、２つの４元ベクトルの内積を求めることが可能となるプロセッサが示されている。内積を求めるための命令の実行時には、１サイクルに７つの演算を行う。

【０００４】

また、特開平１０－１２４４８４においても、４つの乗算器に８つの浮動小数点数を与え、その乗算結果を４入力加算器で並列的に加算し、１回の並列的な乗算と加算とによって内積を求めることが可能なデータプロセッサが示されている。

【０００５】

【発明が解決しようとする課題】

本発明者は、マルチメディア機器において、浮動小数点数を用いたグラフィック処理を従来以上に高速化することが可能なデータプロセッサ及び、データ処理システムについて検討を行った。

【０００６】

マルチメディア機器向けデータプロセッサ、及びマルチメディア処理を行うデータ処理システムにおいて、重要かつ負荷の重い処理は、三次元グラフィクス処理と画像処理である。このうち、画像処理については規格化がなされているため、最も製造コストのかからない実現方法は専用ハードウェアを搭載することである。従来のプロセッサにおいても画像処理用の専用ハードウェアを搭載しているものは既に存在している。

【０００７】

一方、三次元グラフィクス処理においては、座標計算等のジオメトリ処理と配色等のレンダリング処理とが必要となる。レンダリング処理については、汎用プロセッサ向きでなく、定型的な処理が一般的なため、高速処理が必要ならば専用ハードウェアを用いるのが一般的である。これに対して、座標計算等のジオメトリ処理については、自由度が高く浮動小数点データを扱うため、プロセッサの浮動小数点ユニットで処理するのが一般的である。このジオメトリ処理において最も頻繁に行われるのが4元ベクトル内積演算である。輝度計算は内積、座標変換は４×４行列と４元ベクトルの積、変換行列生成は４×４行列同士の積を求めることで処理される。それらの処理は、輝度計算が１回の４元ベクトル内積演算、座標変換が４回の４元ベクトル内積演算、変換行列生成については１６回の４元ベクトル内積演算を行うことによって求めることができる。従来のプロセッサにおいて、４元ベクトル内積演算に特化して高速化を図り、効率的にジオメトリ処理の高速化を達成したものも存在している。

【０００８】

しかし、三次元グラフィクス処理の高速化要求は極めて強く、動画像のリアリティを増すためには、更なる高速化が必要である。しかしながら、グラフィクス処理の基本データは４元ベクトルであるため、従来のプロセッサの方式では並列度をこれ以上上げることは困難である。ＦＩＲ（Finite Impulse Response）フィルタ処理のように、多元ベクトル内積命令を定義すれば高速化されるアプリケーションも多いが、コンシューママルチメディア分野で最も高い浮動小数点演算性能が要求されるのは三次元グラフィクス処理である。従来知られているベクト

ル内積命令を有するプロセッサが効率的に並列度を向上したとしても、三次元グラフィクス処理の高速化に寄与しなければ意味がないものとなる。

【０００９】

一方、ＳＩＭＤ方式では、原理的には並列度を上げることは容易である。しかし、ＳＩＭＤ方式は非効率な面も有しており、並列度を上げることによるコストは極めて増大する傾向にある。従来存在しているプロセッサにおいても既に大きな面積を割いているＳＩＭＤ部分を更に何倍にもすることは現実的な解決方法とは言えない。例えば、従来の技術で示した第１の文献のデータプロセッサでは、上述した性能を実現するために、実際に１０個の浮動小数点積和演算器を内蔵しており、チップ面積は０．２５μｍプロセスで製作したとしても、２４０平方ミリメートルと巨大な面積を必要とする。このうち、４つの浮動小数点積和演算を実行する並列ＳＩＭＤ型浮動小数点ユニットの面積をチップ写真から推定すると２２平方ミリメートル程度となる。除算器を完全には４並列ＳＩＭＤ化していないことと、制御回路は必ずしも4倍にならないこと等により、通常の浮動小数点ユニットの面積の約３倍程度必要となる。

【００１０】

また、従来の技術で示した第２の文献で示されているデータプロセッサのチップ面積は、０．２５μｍプロセスで製作したとすれば５６平方ミリメートル程度となる。このうち、浮動小数点ユニットの面積をチップ写真から推定すると、１０平方ミリメートル程度となる。そして、内積命令用演算器を除いた面積は約７．５平方ミリメートル程度である。したがって、内積命令追加により浮動小数点ユニットの面積が約１．３倍になっていることとなる。

【００１１】

本発明の目的は、演算並列度を効率的向上したデータプロセッサ及びデータ処理システムを提供することである。

【００１２】

本発明の別の目的は、回路規模の増大を極力抑え、浮動小数点数の演算を高精度かつ高速に処理することが可能なデータプロセッサ及びデータ処理システムを提供することである。

【００１３】

【課題を解決するための手段】

本願において開示される発明のうち、代表的なものの概要を簡単に説明すれば下記の通りである。

【００１４】

すなわち、データプロセッサは、浮動小数点数の演算処理能力を向上させるため、ＳＩＭＤ型演算器を内部に構成し、前記ＳＩＭＤ型演算器にベクトルデータの処理を行わせるための単一の命令を有している。また、データプロセッサは、浮動小数点数の演算処理能力を向上させるため、ＳＩＭＤ型演算器を有し、前記ＳＩＭＤ型演算器にベクトルデータの演算を行わせるための命令が、命令セットの中に含まれているものである。更に、前記ＳＩＭＤ型演算器は、浮動小数点数の積和演算を行う複数の演算器を有しているものである。

【００１５】

また、データプロセッサの命令セットには、前記データプロセッサにベクトルデータの内積とスカラーデータとの和を演算させるための命令が含まれている。その命令により、データプロセッサは、例えば４元ベクトルと４元ベクトルとの内積と、前記内積とスカラーデータとの和を一つの命令により求めることを可能とするものである。前記演算を行うため、データプロセッサは、浮動小数点演算器を有しているものである。浮動小数点演算器は、処理能力を高めるためにＳＩＭＤ型演算器とすることも出来る。

【００１６】

前記ＳＩＭＤ型演算器を構成している演算器或いは浮動小数点演算器は、ベクトルの内積とスカラーデータとの和を高速に演算するため、多入力加算器を有しているものである。３次元グラフィックス処理を高速に行うため、３次元グラフィックス処理において多用される４元ベクトル処理に特化したデータプロセッサにおいては、演算器は９入力加算器を有するものである。

【００１７】

また、データプロセッサの命令セットには、一つの命令によって前記データプロセッサに行列データとベクトルデータとの積を演算させるものである。この命

令により、データプロセッサは、４×４行列と４元ベクトルとの積を一つの命令で演算することが可能となる。データプロセッサは、前記命令を処理するため、ベクトルデータとベクトルデータとの内積を求める浮動小数点演算器を複数個有しているものである。３次元グラフィックス処理において多用される４×４行列と４元ベクトルとを用いた演算処理を高速に行うことが可能となる。前記浮動小数点演算器のそれぞれは、内積とスカラーデータとの和とを求めることも可能なものである。また、前記演算器は、多入力加算器を有しているものである。

【００１８】

本発明の前記並びにその他の目的と新規な特徴等については、本願発明の明細書の記述及び添付の図面より明らかになるであろう。

【００１９】

【発明の実施の形態】

図１は本発明を適用したデータプロセッサＤＰの構成図である。本実施例のデータプロセッサＤＰは、整数を処理する能力を有する整数ユニットとしての中央処理装置ＣＰＵ、浮動小数点数の演算を行うＳＩＭＤ型浮動小数点ユニット（Single Instruction Multiple Data Floating-point Unit）ＦＰＵ、命令キャッシュＩＣＡ、データキャッシュＤＣＡ、バスコントローラＢＳＣ、複数の周辺モジュールＰＭ、アドレス端子ＡＴ、データ端子ＤＴ等を有している。中央処理装置ＣＰＵとＳＩＭＤ型浮動小数点ユニットＦＰＵとは、命令バスＩＢを介して命令キャッシュＩＣＡに接続されており、命令キャッシュより命令を取り込む。中央処理装置とＳＩＭＤ型浮動小数点ユニットとが取り込む命令を指定するためのアドレスは、命令アドレスバスＩＡを介して中央処理装置より与えられる。中央処理装置とＳＩＭＤ型浮動小数点ユニットとデータキャッシュＤＣＡとは、データバスＤＢに接続されている。データキャッシュには、データアドレスバスＤＡを介して中央処理装置よりデータアドレスが供給される。命令キャッシュＩＣＡとデータキャッシュＤＣＡとは、キャッシュコントローラを有しているが、図示は省略している。命令キャッシュ及びデータキャッシュは、データ信号やコントロール信号も伝達されるキャッシュバスＣＢを介してバスコントローラＢＳＣに接続されている。命令キャッシュにおいて、キャッシュミス等が生じた場合、外部

アクセスのための命令アドレスは外部アクセス命令アドレスバスＥＩＡを介してバスコントローラに与えられる。また、データキャッシュにおいて、キャッシュミス等が生じた場合、外部アクセスのためのデータアドレスは外部アクセスデータアドレスバスＥＤＡを介してバスコントローラに与えられる。バスコントローラは、キャッシュより送られた命令アドレス又はデータアドレスに従って、アドレス端子ＡＴ及びデータ端子ＤＴを介して接続される外部メモリなどをアクセスするために、外部バスサイクルを起動する。その後、バスコントローラは、外部メモリなどよりデータ端子に到着した命令或いはデータを、キャッシュバスＣＢを介して命令キャッシュ或いはデータキャッシュに供給する。また、バスコントローラには、特に制限されないが、タイマやシリアルコミュニケーションインタフェースコントローラ等の周辺回路ＰＭが周辺バスＰＢを介して接続されている。尚、本実施例のデータプロセッサは、特に制限されないが、単結晶シリコンのような一つの半導体基板上に形成されている。また、特に制限される訳ではないが、本実施例のデータプロセッサは、ＲＩＳＣ（Reduced Instruction Set Computer）アーキテクチャを有し、命令セットの中には浮動小数点命令を有する。浮動小数点命令は、メモリ効率を高めるために１６ビット長であっても良く、命令数が多くなっても対応できるように３２ビット長であってもよいが、それらに限定される訳ではない。

【００２０】

本実施例のデータプロセッサをマルチメディア機器、例えばビデオゲーム機器などへ組み込むことで、３次元グラフィックスを充分にサポートしたデータ処理システムを実現することが可能となる。

【００２１】

次に、図１に示したデータプロセッサＤＰについて詳細な説明を行う。本実施例のデータプロセッサは、面積を節約するために、ＳＩＭＤ型浮動小数点ユニットＦＰＵはメモリアドレシング能力を持っていない。つまり、中央処理装置ＣＰＵはＳＩＭＤ型浮動小数点ユニットに代わってメモリアドレシング機能を有することとなる。そのため、中央処理装置は、ＳＩＭＤ型浮動小数点ユニットのためにメモリからデータのフェッチを行うだけでなく、ＳＩＭＤ型浮動小数点ユニッ

トのために浮動小数点命令を含む全ての命令をメモリからフェッチする。中央処理装置によってフェッチされた命令は、命令バスＩＢを介して中央処理装置とＳＩＭＤ型浮動小数点ユニットとの双方に取り込まれてデコードされる。中央処理装置は、デコードした命令がＣＰＵ命令であった場合にはデコードした命令に従って整数処理を実行し、デコードした命令が浮動小数点命令であった場合にはＳＩＭＤ型浮動小数点ユニットに代わってアドレシング処理などを行う。ＳＩＭＤ型浮動小数点ユニットは、デコードした命令がＣＰＵ命令であった場合には命令を無視し、デコードした命令が浮動小数点命令であった場合にはデコードした命令に従って浮動小数点演算を行う。ここで、デコードした命令がロード又はストア命令である場合、中央処理装置はデータアドレスをデータキャッシュに出力してデータのロード又はストアを要求する。その要求に対してデータキャッシュＤＣＡは、入力されたデータアドレスのデータをデータバスＤＢへロード又はデータバスからストアする。尚、ストアの場合も通常のコピーバックキャッシュの外部アクセスはリードであり、リードによってキャッシングされたキャッシュラインの一部にストア動作を行う。但し、キャッシングの際に有効かつ更新されたキャッシュラインがリプレースされた場合は、キャッシュラインを外部にコピーバックする。ロード又はストア命令の対象レジスタがＳＩＭＤ型浮動小数点ユニットのレジスタであった場合、ＳＩＭＤ型浮動小数点ユニットはロード命令ならばデータバス上の値をＳＩＭＤ型浮動小数点ユニットのレジスタに書き込み、ストア命令ならばデータバス上にＳＩＭＤ型浮動小数点ユニット内のレジスタの値を出力する。尚、ＳＩＭＤ型浮動小数点ユニットによるロード及びストアに関してはで図３を用いて詳述する。

【００２２】

図２は図１のデータプロセッサＤＰのＳＩＭＤ型浮動小数点ユニットＦＰＵの構成図である。図中の命令デコーダＩＤＥＣは、命令バスＩＢから供給される命令をデコードし、デコード結果に基づき制御信号ＣＴＲＬを生成し、4つのベクトル浮動小数点ユニットＶ－ＦＰＵを制御する。各ベクトル浮動小数点ユニットＶ－ＦＰＵは１２８ビット幅を有するバスを介してデータバスＤＢに接続されている。それぞれの浮動小数点ユニットは、制御信号ＣＴＲＬに従いロード/スト

ア及び浮動小数点演算の処理を実行する。データバスＤＢのバス幅については、１２８ビット、２５６ビット或いは５１２ビット等で構成することが可能である。５１２ビット幅にすると配線量は大きくなるが、各ベクトル浮動小数点ユニットが専用の１２８ビットバスを持てるため制御が簡単になる。但し、５１２ビット幅のデータバスを有効活用するにはデータキャッシュＤＣＡを５１２ビット幅または１２８ビット幅の４バンクといった構成にする必要がある。データバスが１２８ビット幅の場合、データバスのビット分割または時分割が必要である。例えば、各ベクトル浮動小数点ユニットが３２ビットアクセスをしてＳＩＭＤ型浮動小数点ユニット全体で１２８ビットアクセスする場合はビット分割で対応する。一方、各ベクトル浮動小数点ユニットが１２８ビットアクセスをする場合は時分割が必要である。また、あるベクトル浮動小数点ユニットが１２８ビットアクセスを行い、他のベクトル浮動小数点ユニットについては動作行わせないためのＮＯＰ（non-operation）命令を定義する方式もある。

【００２３】

図3は図2で示したベクトル浮動小数点ユニットＶ－ＦＰＵの一つについて示した構成図である。浮動小数点演算命令実行時には３２ビット4バンクレジスタファイルＲＧＳＴの２つの４バンクリードポートＸ、Ｙと１つの通常リードポートＺからレジスタ値を演算ブロックＥＢＬＫへ転送し、演算ブロックでの演算処理を行う。演算ブロックでの演算処理後、演算用ライトポートＶからレジスタファイルに演算結果を書き込む。ロード命令実行時には、転送ブロックＴＢＬＫはロード制御信号ＬＤＣによってデータバスＤＢにのせられたロードすべきデータを選択し、レジスタファイルの転送用４バンクライトポートＵを介してレジスタファイルに書き込む。データの転送幅が１２８ビット未満の場合、ライトアライナＷＡＬＮはデータを適宜アライメントしてＵポートに送る。ストア命令実行の場合、転送ブロックＴＢＬＫはレジスタファイルＲＧＳＴの転送用４バンクリードポートＷからストアデータを読み出し、バスドライブ信号ＢＤＣに基づいて読み出したデータをデータバスに転送する。この場合も、転送幅が１２８ビット未満の場合はリードアライナＲＡＬＮによって適宜アライメントしてデータバスに送る。また、レジスタ間転送命令実行時には、転送ブロックは、リードポートＷか

らレジスタ値を読み出し、ロード制御信号ＬＤＣによってリードポートWより読み出されたデータ選択し、ライトポートUからレジスタファイル書き込む。この場合も、転送幅が１２８ビット未満の場合、リードアライナまたはライトアライナで適宜アライメントを行いレジスタ間転送を行う。

【００２４】

尚、レジスタの構成方法として、３２ビット幅のレジスタファイルをレジスタ番号で４バンクに分割して定義する方法と、１２８ビット幅のレジスタをビット方向に４分割して定義する方法との２つが考えられる。前者の利点は３２ビット幅で４元ベクトルを定義できることであり、後者の利点は１本のレジスタで４元ベクトルを定義できることである。しかし、後者の方法では、データがベクトルかスカラかによらず1本のレジスタに収まるため、レジスタリード･ライトやフォワーディング処理が容易ではあるが、レジスタの使用効率や拡張性が悪くなる。例えば、１２８ビット幅のレジスタに対して内積命令を定義した場合、演算出力を格納するレジスタについても１２８ビットであるため、上位９６ビットが無駄になってしまう。また、内積命令の定義に１２８ビット幅を使用してしまうと、内積命令のＳＩＭＤ化が困難となる。例えば４並列にしようとすると５１２ビット幅が必要となる。一方、前者の方式では３２ビット幅で４元ベクトル内積命令が定義できるため、出力レジスタの無駄はなくＳＩＭＤ化が容易である。以上より、レジスタファイルをレジスタ番号で４バンクに分割して４元ベクトルを定義し、その４元ベクトルに対して内積命令を定義し、レジスタファイルと内積命令とをＳＩＭＤ化して並列度を上げることが可能であり、効率的に１命令あたりの並列度を上げることができる。本実施形態では前者の方法を採用することとする。

【００２５】

図４は図３の３２ビット４バンクレジスタファイルＲＧＳＴの構成図である。４つあるバンクのそれぞれのバンクＢＡＮＫ０、ＢＡＮＫ１、ＢＡＮＫ２、ＢＡＮＫ３は、４リード、２ライト、３２ビット、１６本のレジスタファイルから構成される。つまり、レジスタ数は４バンク×１６本で、合計６４本となる。レジスタファイルは、６ビットで規定されるレジスタ番号の上位４ビットと、レジス

タライト制御装置であるＷＣＵとＷＣＶからのライトイネーブル信号、ＷＥＵ［０］、ＷＥＵ［１］、ＷＥＵ［２］ＷＥＵ［３］、ＷＥＶ［０］、ＷＥＶ［１］、ＷＥＶ［２］ＷＥＶ［３］によって制御されている。

【００２６】

レジスタライト制御装置であるＷＣＵは、ライトレジスタ番号ＷＮＵの下位2ビット４２とライトサイズＷＳＩＺＥとから、ライトするバンクを決定し、更にライト指示ＷＲＩＴＵがアサートされたら、ライトすべきバンクのライトイネーブル信号ＷＥＵをアサートする。ライトサイズＷＳＩＺＥが１２８ビットの場合は、ライトレジスタ番号ＷＮＵに関係なく全てのバンクのライトイネーブル信号をアサートする。ライトサイズが６４ビットの場合は、ライトレジスタ番号ＷＮＵに応じてバンク０、１または２、３のライトイネーブル信号をアサートする。ライトサイズが３２ビットの場合は、ライトレジスタ番号ＷＮＵに応じて４つのバンクのうち、１つのバンクを指定するためのライトイネーブル信号をアサートする。以上により、転送用４バンクライトポートＵを介してレジスタにデータが書き込まれる。演算用ライトポートＶを介してレジスタにデータを書き込む場合、演算用ライトポートは常に３２ビットライトなので、レジスタライト制御装置ＷＣＶはライト指示ＷＲＩＴＶがアサートされたら、ライトレジスタ番号ＷＮＶの下位2ビット４２に応じて４バンクのうち、１つのバンクを指定するためのライトイネーブル信号ＷＥＶをアサートする。

【００２７】

各バンクでは、リードレジスタ番号ＲＮＷ、ＲＮＸ、ＲＮＹ、ＲＮＺの上位4ビット４１によって、１６本の内の１本の３２ビットレジスタが指定される。本実施例では、バンクが４つあるため、合計４本のレジスタをリードして出力することが可能である。転送用４バンクリードポートＷは、転送幅が１２８ビット未満の場合は転送ブロックＴＢＬＫがアライメントするので、各バンクの出力を直接出力する。４バンクリードポートＸ、Ｙポートは、ベクトル命令用の１２８ビット出力時は転送用４バンクリードポート同様、各バンクの出力を直接出力する。また、通常演算用の３２ビット出力時にはレジスタ番号ＲＮＷ、ＲＮＸ、ＲＮＹ、ＲＮＺの下位2ビット４２を用い、セレクタ４３２と４３３とにより、リー

ドすべきバンクを選択しＸ［０］とＹ［０］とに出力する。通常リードポートＺは、通常の３２ビットポートなので、常にレジスタ番号ＲＮＷ、ＲＮＸ、ＲＮＹ、ＲＮＺの下位2ビット４２を用い、セレクタ４３１により、リードすべきバンクを選択して出力する。この結果、Ｘ［０］、Ｙ［０］、Ｚには６４本のレジスタの任意の3本の値を載せることができる。

【００２８】

なお、本実施例のレジスタファイルＲＧＳＴは、転送系の転送用４バンクライトポートＵ、転送用４バンクリードポートＷと、演算系の演算用ライトポートＶ、４バンクリードポートＸ、Ｙ、通常リードポートＺとにポートが分かれているため、スーパスカラやＶＬＩＷ（Very Long Instruction Word）アーキテクチャを適用することも可能である。

【００２９】

図5は図3の演算ブロックＥＢＬＫの構成図である。本実施例の演算ブロックは２つの浮動小数点４元ベクトルの内積と浮動小数点数との和を計算する。即ち、Ｘ［０］×Ｙ［０］＋Ｘ［１］×Ｙ［１］＋Ｘ［２］×Ｙ［２］＋Ｘ［３］×Ｙ［３］＋Ｚを計算する。符号処理部ＳＰＰ、指数処理部ＥＰＰ、４つの乗算部（ＭＬＰ０、ＭＬＰ１、ＭＬＰ２、ＭＬＰ３）、４つのアライナ（ＡＬＮ０、ＡＬＮ１、ＡＬＮ２、ＡＬＮ３）、ＺアライナＡＬＮＺ、９入力加算器ＡＤＤＲ、正規化部ＮＯＲから成る。正規化部ＮＯＲは、正規化以外に、正数化や丸め処理も行う。

【００３０】

入力される各Ｘ及び各Ｙの浮動小数点数の符号部は符号処理部ＳＰＰへ、指数部は指数処理部ＥＰＰへ、仮数部は乗算部（ＭＬＰ０、ＭＬＰ１、ＭＬＰ２、ＭＬＰ３）へ入力される。

【００３１】

符号処理部ＳＰＰは、Ｘ及びＹの符号が入力され、ＸＸＹ4つの積の符号Ｓ［０］、Ｓ［１］、Ｓ［２］、Ｓ［３］をＥＯＲゲートによって生成する。更に、その結果とＺの符号ＳＺとのＥＯＲを取ることにより、４つの積がＺと異符号かどうかをチェックし、チェック結果をＩｎｖ［０］、Ｉｎｖ［１］、Ｉｎｖ［２

]、Ｉｎｖ［３］として各アライナＡＬＮ０、ＡＬＮ１、ＡＮＬ２、ＡＬＮ３及び９入力加算器ＡＤＤＲに出力する。異符号の基準としたＳＺは正数化前の符号となるので正規化部ＮＯＲに送られる。

【００３２】

指数処理部ＥＰＰは、ＸＸＹの４つの積の指数部とＺの指数部との５つ項の中での最大指数Ｅｍａｘを求めて正規化部に送出する。更に、Ｅｍａｘと各項の指数差を求め、Ｅｄｉｆｆ［０］、Ｅｄｉｆｆ［１］、Ｅｄｉｆｆ［２］、Ｅｄｉｆｆ［３］、ＥｄｉｆｆＺとして５つのアライナＡＬＮ０、ＡＬＮ１、ＡＬＮ２、ＡＬＮ３、ＡＬＮＺに出力する。詳細については図６～１０で説明する。

【００３３】

各乗算部ＭＬＰ０、ＭＬＰ１、ＭＬＰ２、ＭＬＰ３には、各Ｘ及び各Ｙの仮数部が入力される。入力されたＸの仮数部及びＹの仮数部の積をキャリー保存形式で求め、ＭＣ［０］とＭＳ［０］、ＭＣ［１］とＭＳ［１］、ＭＣ［２］とＭＳ［２］、ＭＣ［３］とＭＳ［３］のペアとしてそれぞれ対応するアライナＡＬＮ０、ＡＬＮ１、ＡＬＮ２、ＡＬＮ３に出力する。キャリー保存形式とは、キャリー伝播加算器で加算すると通常の2進数となる形式で、キャリー伝播なしで生成することができるため高速に生成できる。

【００３４】

アライナＡＬＮ０、ＡＬＮ１、ＡＬＮ２、ＡＬＮ３は、符号処理部ＳＰＰからの異符号Ｉｎｖ、及び指数処理部ＥＰＰからの指数差Ｅｄｉｆｆによって、乗算部ＭＬＰ０、ＭＬＰ１、ＭＬＰ２、ＭＬＰ３から出力されるキャリー保存形式の仮数部の積のアライメント及び論理反転を行う。各アライナの出力は、ＭＣａｌｎ［０］とＭＳａｌｎ［０］、ＭＣａｌｎ［１］とＭＳａｌｎ［１］、ＭＣａｌｎ［２］とＭＳａｌｎ［２］、ＭＣａｌｎ［３］とＭＳａｌｎ［３］のペアとして9入力加算器ＡＤＤＲに送られる。詳細は図１０で説明する。Ｚアライナは、指数処理部からの指数差ＥｄｉｆｆＺによって、Ｚの仮数部ＭＺのアライメントを行い、ＭＺａｌｎとして9入力加算器に送る。詳細は図１１で説明する。

【００３５】

9入力加算器は、ＭＣａｌｎ［０］、ＭＳａｌｎ［０］、ＭＣａｌｎ［１］、Ｍ

Ｓａｌｎ［1］、ＭＣａｌｎ［2］、ＭＳａｌｎ［2］、ＭＣａｌｎ［3］、ＭＳａｌｎ［3］、及びＭＺａｌｎを加算し、結果をＭａｃｍとして正規化部に送出する。異符号の項の加算のために、４つのアライナＡＬＮ０、ＡＬＮ１、ＡＬＮ２、ＡＬＮ３で４つのキャリー保存形式ペアの論理反転が行われるが、符号反転のためには更に＋１する必要がある。＋１は9入力加算器へのキャリーインで行うため、Ｉｎｖ［0］、Ｉｎｖ［1］、Ｉｎｖ［2］、Ｉｎｖ［3］を入力してキャリーイン数を制御している。詳細は図１２、１４を用いて説明する。

【００３６】

正規化部ＮＯＲは、符号生成部からのＳＺ、指数処理部からのＥｍａｘ、9入力加算器からのＭａｃｍを受け取り、通常の浮動小数点演算器と同様に正規化、正数化及び丸めを行い、最終演算結果Ｖを生成してレジスタファイルＲＧＳＴに送る。

【００３７】

図６は、指数処理部ＥＰＰについての説明である。この例では、高速化のために指数差を4入力加算器で直接求めている。一般的には、4つの積の指数部を求め、これとＺの指数部の中から最大指数部を求め、さらに各指数部と最大指数部の差をとって指数差を求める手法が考えられるが、低速である上に、４つの加算器、４～１０個の大小比較器（比較の並列度によって異なる）、5つの減算器が必要となり、論理規模が削減される訳ではない。本実施形態の指数処理部は、図６に示したように、指数差生成部ＥＤＧと出力選択部ＥＯＳとから成る。

【００３８】

図７は、指数差生成部ＥＤＧの構成である。5項から選び得る全ての2項の指数差を１０個の４入力加算器ＦＡＤＲで求めている。例えば、Ｘ［0］×Ｙ［0］とＸ［1］×Ｙ［1］の指数差は、ＥＸ［0］＋ＥＹ［0］－ＥＸ［1］－ＥＹ［1］である。各指数部は、規格に基づくバイアス“１２７”がかかっている。しかし、上式では４つのバイアスが打ち消し合うので上式で求めた指数差のバイアスは“0”である。一方、Ｘ［3］×Ｙ［3］とＺの指数差は、ＥＸ［3］＋ＥＹ［3］－ＥＺ－１２７である。バイアスが打ち消されるように－１２７を行う。

【００３９】

図８は図７の4入力加算器ＦＡＤＲの詳細を示した図である。４つの入力ＩＰＴ０、ＩＰＴ１、ＩＰＴ２、ＩＰＴ３とが入力される8ビット4入力キャリー保存加算器ＦＡＤＲＳと、9ビットキャリー伝播加算器ＦＡＤＲＰから成る。図7では4項のうち2項を論理反転しただけなので、この2項の符号反転のために、2ビットのキャリーインがある。指数差の範囲を－５１０～５１０で表現するには符号付き２進数で１０ビットが必要なため、入力を１０ビットまで符号拡張して１０ビット幅で計算するのが単純である。しかし、実際には上位ビットは冗長であるため図８に示した構成とする。本実施形態の構成では、ビットキャリー伝播加算器ＦＡＤＲＰからのキャリーアウトは4入力加算結果が正の場合に１となる。したがって、キャリーアウトは符号反転しなかった指数の方が大きいか２つの指数が等しいことを表わしている。このキャリーアウトをＧＥ信号として出力し、指数部の大小判定に使用する。尚、８ビット４入力キャリー保存加算器は図１５のような１ビット４入力キャリー保存加算器を８個並べることにより構成される。

【００４０】

図９は図６の出力選択部ＥＯＳの詳細を示した図である。まず、６２０において、指数差生成部ＥＤＧから出力される１０本のＧＥ信号（ＧＥ０１、ＧＥ０２、ＧＥ０３、ＧＥ１２、ＧＥ１３、ＧＥ０Ｚ、ＧＥ１Ｚ、ＧＥ２Ｚ、ＧＥ３Ｚ）に基づき、選択制御信号ｓｅｌ０、ｓｅｌ１、ｓｅｌ２、ｓｅｌ３、ｓｅｌｚを生成する。ｓｅｌ０、ｓｅｌ１、ｓｅｌ２、ｓｅｌ３、ｓｅｌｚのそれぞれは、Ｘ［０］×Ｙ［０］、Ｘ［１］×Ｙ［１］、Ｘ［２］×Ｙ［２］、Ｘ［３］×Ｙ［３］、Ｚが最大指数を持つことを表わしている。但し、指数部が等しい場合は、この順に優先度があり、選択制御信号はホットワンであることが保証されている。例えば、全ての指数部が等しい場合はｓｅ１０のみがアサートされる。次に、６３０において、上記で生成された選択制御信号を基に最大指数Ｅｍａｘを生成する。4入力加算器ＦＡＤＲで指数差を直接生成したため、最大指数は６３０において改めて生成することとなる。図のように、６３１及び６３２において、選択制御信号で指数部を選択し、8ビットキャリー伝播加算器６３３で加算する。尚、ＥＺを選択した場合は、他の場合とバイアスを合わせるためにＥＺ＋１２

７を計算する。次に、６４０～６４４において、各項の指数差Ｅｄｉｆｆ［０］、Ｅｄｉｆｆ［１］、Ｅｄｉｆｆ［２］、Ｅｄｉｆｆ［３］、ＥｄｉｆｆＺを求める。Ｅｍａｘを求める前に指数差を計算しているため、指数差が「Ｅｍａｘ－各項の指数」となっていないため、符号反転しなければならない場合が存在する。このため、指数差生成部ＥＤＧから出力される指数差の一部については、セレクタ入力前に論理反転させている。更に、自分自身が最大指数であった場合は指数差は“０”なので、“０”を入力している。また、符号反転を完結させるには論理反転後に＋１する必要がある。そこで、＋１が必要であることを示す信号Ｅｄｉｆｆｐ１［０］、Ｅｄｉｆｆｐ１［１］、Ｅｄｉｆｆｐ１［２］、Ｅｄｉｆｆｐ１［３］、Ｅｄｉｆｆｐ１Ｚを、６５０によって求める。尚、本実施例では、指数差に対する＋１は行わず、指数差を使用するアライナにおいて、＋１が必要な場合には＋１の代りに１ビットシフトしている。

【００４１】

図１０において、アライナＡＬＮ０、ＡＬＮ１、ＡＬＮ２、ＡＬＮ３について説明する。本実施例のアライナは、キャリー保存形式の４つの積のアライナである。キャリー側ＭＣ［ｎ］（ｎは０から３までの数）用とサム側ＭＳ［ｎ］用とにアライナが２つあり、同一の信号で制御される。まず、シフタＳＦＴ１、ＳＦＴ２により、指数差Ｅｄｉｆｆ［ｎ］分だけ右シフトを行う。次にＥｄｉｆｆｐ１［ｎ］が１の場合、１ビットシフタＳＦＴ１’、ＳＦＴ２’において、更に１ビットシフトする。そして、６６２及び６７２においては、Ｉｎｖ［ｎ］が１の場合、論理反転を行う。以上により、ＭＣ［ｎ］に対応してはＭＣａｌｎ［ｎ］を出力し、ＭＳ［ｎ］に対応してはＭＳａｌｎ［ｎ］を出力する。

【００４２】

図１１はＺアライナＡＬＮＺである。まず、シフタＳＦＴＺに入力されるＭＺを指数差ＥｄｉｆｆＺだけ右シフトする。次に、１ビットシフタＳＦＴＺ’においてＥｄｉｆｆｐ１Ｚが1の場合、更に1ビットシフト行い、結果をＭＺａｌｎとして出力する。尚、異符号判定をＺを基準に行っているため、Ｚの符号反転は不要である。

【００４３】

図１２においては、図５で示した９入力加算器ＡＤＤＲについて詳細について説明する。本実施例では、９入力加算ではあるがキャリー保存形式によって５入力が９入力になっているため、桁数増加は最大3ビットである。まず、アライナＡＬＮ０、ＡＬＮ１、ＡＬＮ２、ＡＬＮ３、ＡＬＮＺから出力される、９つの入力のそれぞれを符号拡張部ＳＥにおいて３ビット符号拡張する。符号拡張部の出力は、９入力キャリー保存加算器アレイＣＳＡに入力される。次に、符号反転項数に応じたキャリーインを行う。キャリー保存形式の積の１ペアを反転するのに2ビットのキャリーインを行うため、４つの積に対して最大８ビットのキャリーインが必要である。符号反転はＩｎｖ［０］、Ｉｎｖ［１］、Ｉｎｖ［２］、Ｉｎｖ［３］で制御しているので、図のようにこの4信号のそれぞれに2ビットのキャリーを対応させる。そして、９入力キャリー保存加算器アレイＣＳＡに６ビット、キャリー伝播加算器ＣＰＡに２ビットのキャリーインを行う。キャリー伝播加算器は、上記の２ビットのキャリーと、９入力キャリー保存加算器アレイからのキャリー出力ＣＯＵＴと、サム出力ＳＯＵＴとから、正規化前仮数部Ｍａｃｍを生成する。

【００４４】

図１３は、図１２で示した９入力キャリー保存加算器アレイＣＳＡの詳細図である。図中左側から３ビット符号拡張部の出力が入力される。まず、１段目を３入力キャリー保存加算器７３０、７３１、７３２とし、２段目も３入力キャリー保存加算器７３３、７３４とし、3段目を４入力キャリー保存加算器７３５とする。この構成により、項数が９から６、４、２と減少し、最終的にキャリー出力ＣＯＵＴ及びサム出力ＳＯＵＴを得る。また、本実施例では、キャリーインも、ＣＩ０、ＣＩ１、ＣＩ２、ＣＩ３、ＣＩ４、ＣＩ５の6ビットまで可能である。尚、３入力キャリー保存加算器は、例えば、図１４のような１ビット3入力キャリー保存加算器をビット幅だけ並べて構成する。また、４入力キャリー保存加算器は、例えば図１５のような１ビット４入力キャリー保存加算器で構成する。

【００４５】

図１６には、本発明の一つの実施形態であるデータ処理システムのブロックダ

イヤグラムを示している。同図において、上述したデータプロセッサＤＰはバスＢＵＳに接続されている。ここで示したバスは、アドレスが転送されるバスとデータが転送されるバスとを含むものとする。また、バス幅等について制限されるものではない。更に、命令とデータとが同一のバスを転送される構成であっても、命令とデータとが別のバスを介して転送される構成であっても、同図で示されるバスに含まれるものである。それらバス構成は、データ処理システムにおける処理速度、面積効率、或いは接続されるデバイスの構成等によって様々に変更可能である。

【００４６】

上記バスＢＵＳには、データプロセッサの作業領域やデータの一時記憶領域として使用されるＳＲＡＭや、データプロセッサのＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）等が記憶されるＲＯＭが接続されている。また、バスには、制御回路ＤＭＣを介してＤＲＡＭが接続されている。制御回路ＤＭＣは、ＤＲＡＭに対してアドレスマルチプレクス制御やリフレッシュ制御等を行うものとするが、ＤＲＡＭ内部やデータプロセッサ内部に分散させる構成であってもよい。更に、バスには周辺装置制御部ＰＤＣと表示コントローラＤＣとが接続されている。周辺装置制御部には、光ディスク等の外部記憶装置ＥＳＤやキーボードＫＢＤ等が接続されている。また、表示コントローラには、表示装置ＤＰが接続されている。

【００４７】

上記データプロセッサは、浮動小数点演算を実行するため命令を備えており、命令の転送のための浮動小数点演算のためのレジスタを有しているため、３次元グラフィック処理に多用される浮動小数点数の演算を高速で実行することが可能である。従って、マルチメディア機器であるゲーム機や携帯情報端末等として利用される本実施形態のデータ処理システムは、全体のコストを削減し、３次元グラフィック処理を高精度かつ高速に処理することが可能となる。

【００４８】

また、図１６のデータ処理システムにおいて、バスＢＵＳにレンダリングコプロセッサを追加することも可能である。３次元グラフィックス処理は、ジオメトリ処理とレンダリング処理とで構成される。内積演算やベクトル変換演算を多用

するジオメトリについては、本実施形態のデータプロセッサＤＰによって処理させ、レンダリング処理については、レンダリングコプロセッサに処理させる。このことにより、レンダリング処理をデータプロセッサ内の中央処理装置に処理させるデータ処理システムよりも、３次元グラフィックス処理を高速に処理することが可能なデータ処理システムを提供することが可能となる。

【００４９】

以上、本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明は記載している実施形態に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。

【００５０】

例えば、データプロセッサは、メモリマネジメントユニットなど本実施形態で示したブロック以外のブロックを含むことも可能であり、本実施形態で示したブロックの配置等を変更することも可能である。また、データプロセッサは、スーパースカラアーキテクチャを採用することも可能である。スーパースカラアーキテクチャを採用することで、１本のパイプでは演算処理の命令を実行し、別のパイプではデータのメモリからのロードやメモリへのストアを行うことが可能となり、高速処理が可能となる。

【００５１】

また、本実施形態においては、データプロセッサに形成されたＦＰＵについて記載してきた。しかし、本発明の思想はＦＰＵに限定されるものではなく、整数演算ユニットへの適用も可能である。整数演算ユニットへの適用の際、図５の乗算アレイ及び９入力加算器を用いれば整数型の４元ベクトルの内積とスカラの和との演算器を実現することが可能となる。整数型演算器の場合、純粋な１６並列ＳＩＭＤ方式に比べて、演算器の論理規模の大幅削減はできないが、同等の演算並列度を１／４の４並列ＳＩＭＤで達成することが可能となる。また、レジスタの幅も１／４で良いのでレジスタの論理規模を大幅に削減することが可能となる。

【００５２】

以上、本願で開示した本発明の構成及び動作によって、上述した効果及び下記

に示す効果を得ることができる。

【００５３】

本発明のデータプロセッサ及びデータ処理システムは、その命令セットの中に、図１のデータプロセッサに搭載されたＳＩＭＤ－ＦＰＵにおいて４元ベクトルの内積とスカラとの和を浮動小数点形式で4つ処理させるための命令を持たせることが可能となる。上記構成をとり、上記命令を持たせることで、１サイクルで１命令を実行する場合、１サイクルで２８の浮動小数点演算の実行が可能である。

【００５４】

更に、ＳＩＭＤ－ＦＰＵのそれぞれの演算器を、４元ベクトルの内積とスカラーデータとの加算を行うことを可能とする構成とし、更に、データプロセッサの命令セットに内積と加算とを行わせる命令をもたせることで、多元ベクトルデータに対応するためことも可能となる。

【００５５】

上記に示した、演算器の内積と加算の処理、と演算器のＳＩＭＤ化とを組み合わせ、対応する命令を命令セットに加えることで、１サイクルに上記１命令を実行した場合、1サイクルに３２の浮動小数点演算を実行することが可能となる。

【００５６】

上記データプロセッサ、及び上記データプロセッサを用いて構成したデータ処理装置は、従来のデータプロセッサ及びデータ処理システムに対し、大幅に演算並列度を上げることが可能となり、処理速度を向上させることが可能となる。上記データプロセッサは、１サイクルあたり３２ＦＬＯＰＳの処理能力を有することとなる。

【００５７】

また、図５に例示している如く、本発明の演算ブロックでは、仮数部の処理及び最大指数項と各項の指数差の計算に、一度だけキャリー伝播加算を行うように構成している。そのため、演算レイテンシの短縮が容易で高周波数動作に適している。

【００５８】

【発明の効果】

本発明によって４元ベクトル内積命令を４並列のＳＩＭＤ化することにより、１命令で２８演算の実行が可能である。更に、４元ベクトルの内積とスカラの和の演算を定義することにより４元を超える多元ベクトルデータに対応する事が可能となり、１命令で３２演算の実行が可能となる。故に、浮動小数点数の演算を高速で処理可能なデータプロセッサを提供することが可能となり、更に、マルチメディア処理特に３次元のグラフィック処理を高速に処理可能なデータ処理システムの提供が可能となる。

【図面の簡単な説明】

【図１】

本発明を適用したプロセッサの構成図。

【図２】

本発明を適用したプロセッサのＳＩＭＤ－ＦＰＵの構成図。

【図３】

ＳＩＭＤ－ＦＰＵのベクトルＦＰＵの構成図。

【図４】

ベクトルＦＰＵの３２ビット４バンクレジスタファイルの構成図。

【図５】

ベクトルＦＰＵの演算ブロックの構成図。

【図６】

演算ブロックの指数演算部。

【図７】

指数演算部の指数差生成部。

【図８】

指数差生成部の４入力加算器。

【図９】

指数演算部の出力選択部。

【図１０】

演算ブロックのアライナ。

【図１１】

演算ブロックのＺアライナ。

【図１２】

演算ブロックの９入力加算器。

【図１３】

９入力加算器の９入力キャリー保存加算器アレイ。

【図１４】

１ビット３入力加算器の例。

【図１５】

１ビット４入力加算器の例。

【図１６】

本発明のデータプロセッサを使用したデータ処理システム。

【符号の説明】

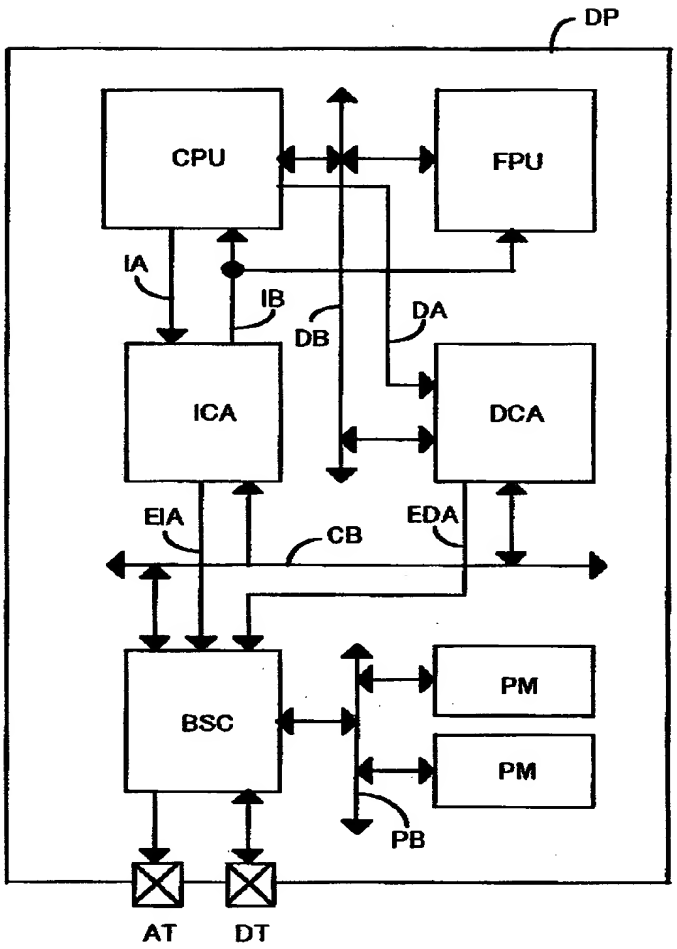
ＤＰ：データプロセッサ、ＣＰＵ：中央処理装置、ＦＰＵ：浮動小数点ユニット、ＩＣＡ：命令キャッシュ、ＤＣＡ：データキャッシュ、ＢＳＣ：バスコントローラ　、ＰＭ：周辺モジュール　、ＩＡ：命令アドレスバス　、ＩＢ：命令バス　、ＤＡ：データアドレスバス　、ＤＢ：データバス　、ＣＢ：キャッシュバス　、ＥＩＡ：外部アクセス命令アドレスバス　、ＥＤＡ：外部アドレスデータアドレスバス　、ＰＢ：周辺バス　、ＡＴ：アドレス端子　ＤＴ：データ端子、Ｖ－ＦＰＵ：ベクトル浮動小数点ユニット、ＩＤＥＣ：命令デコーダ、ＣＴＲＬ：制御信号、ＥＢＬＫ：演算ブロック、ＴＢＬＫ：転送ブロック、ＲＧＳＴ：レジスタファイル、ＢＤＣ：バスドライブ信号、ＷＡＬＮ：ライトアライナ、ＬＤＣ：ロード制御信号、ＲＡＬＮ：リードアライナ、Ｗ（０）：リードポート、Ｖ：演算用ライトポート、Ｘ（０）：リードポート、Ｙ（０）：リードポート、Ｕ：ライトポート、Ｚ：リードポート、ＷＣＵ：レジスタライト制御装置、ＡＤＤＲ：９入力加算器、ＳＰＰ：符号処理部、ＥＰＰ：指数処理部、ＮＯＲ：正規化部

。

【書類名】図面

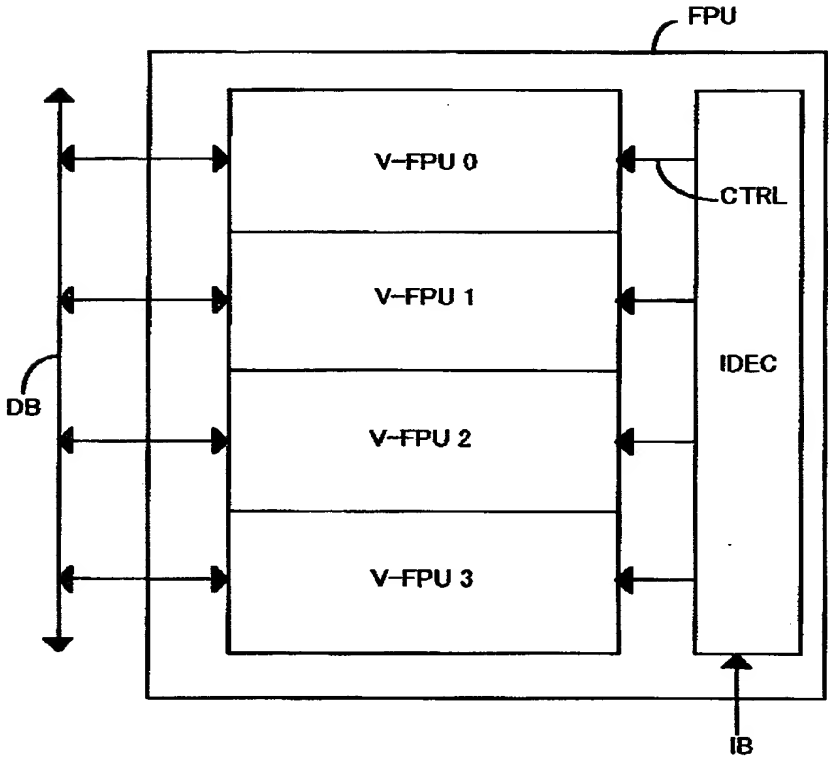
【図１】

図1



【図２】

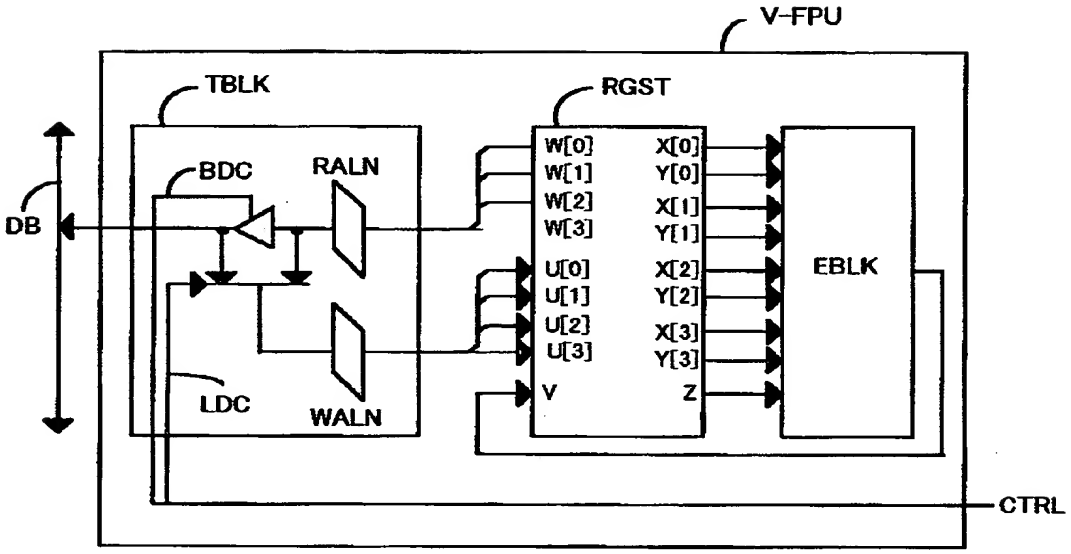
図2



出証特２０００－３０８１６６８

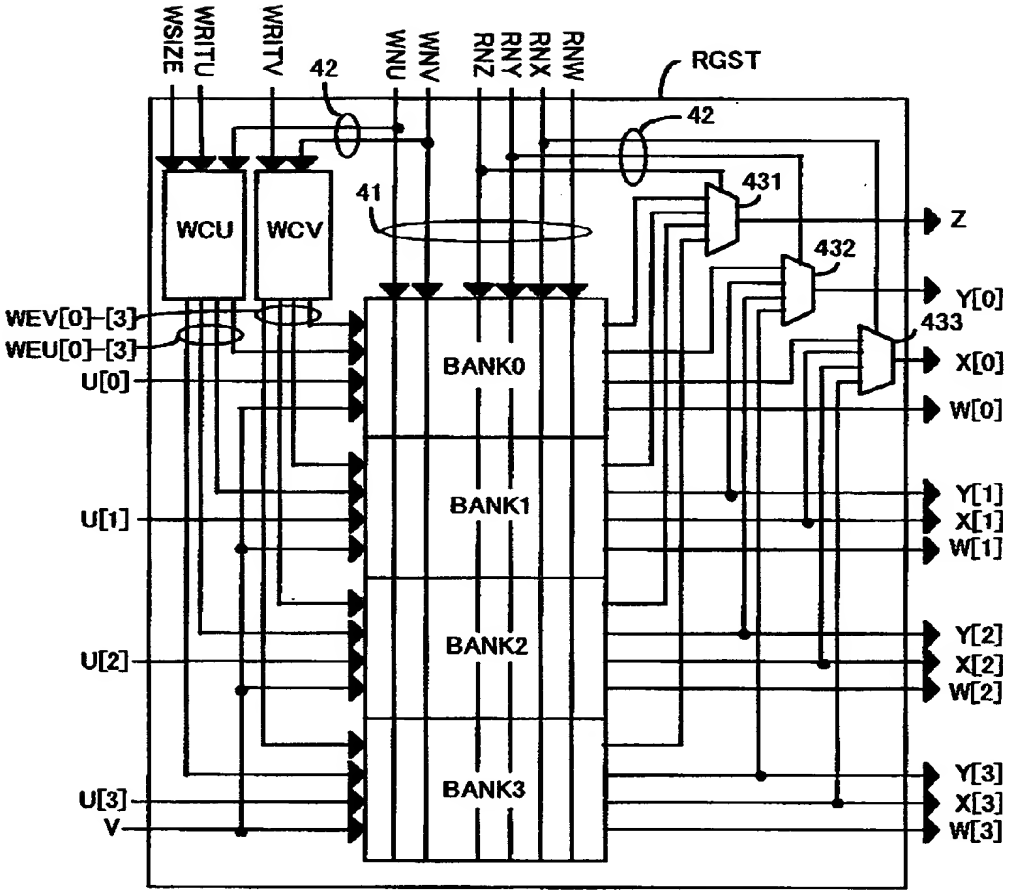
【図３】

図3


V-FPU
TBLK
RGST
BDC
RALN
DB
W[0]
W[1]
W[2]
W[3]
U[0]
U[1]
U[2]
U[3]
V
X[0]
Y[0]
X[1]
Y[1]
X[2]
Y[2]
X[3]
Y[3]
Z
EBLK
LDC
WALN
CTRL

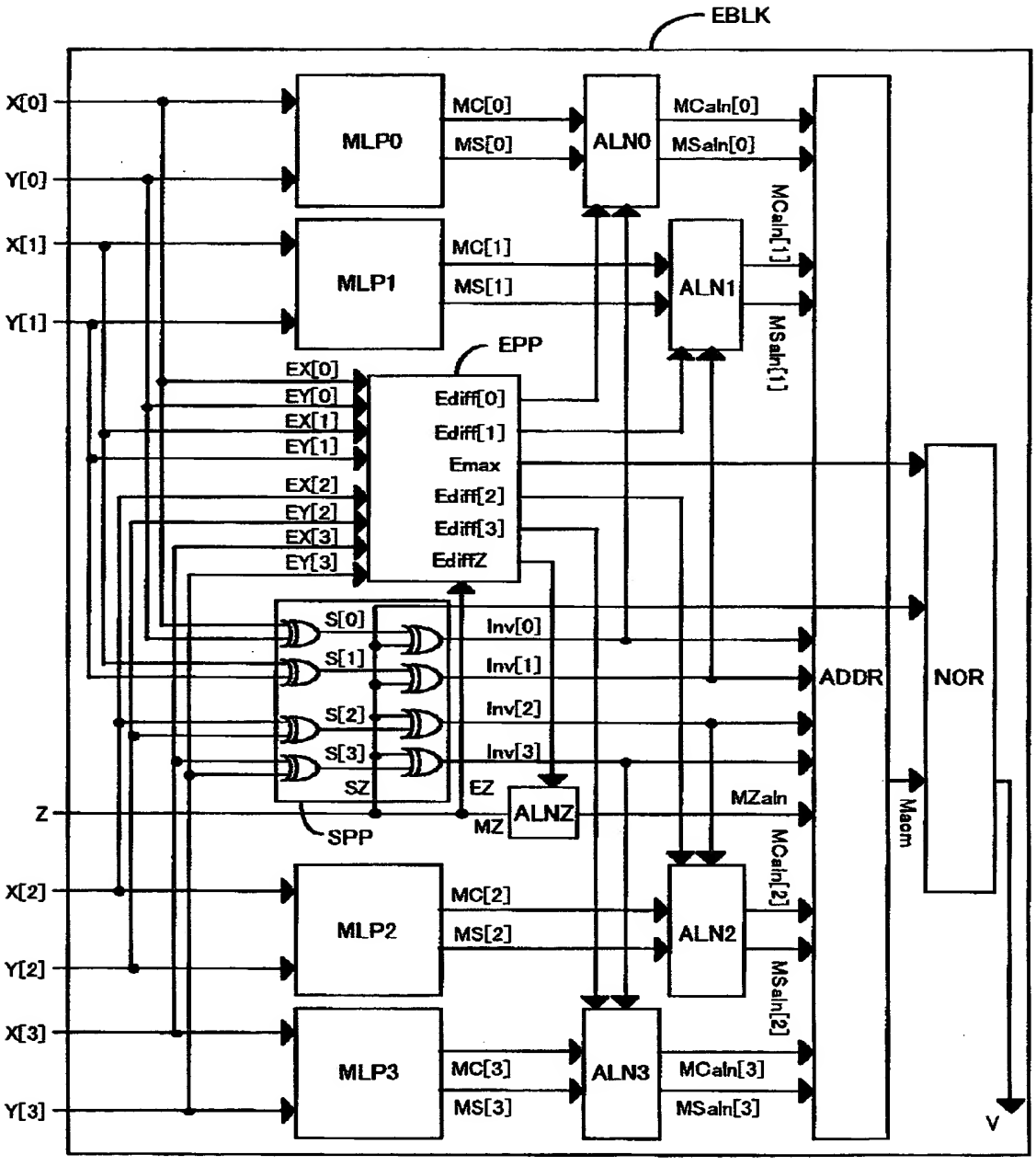
【図４】

図4


WSIZE
WRITU
WRITV
42
WNU
WNV
RNZ
RNY
RNX
RNW
RGST
42
431
41
WCU
WCV
Z
432
Y[0]
433
X[0]
WEV[0]-[3]
WEU[0]-[3]
U[0]
BANK0
W[0]
U[1]
BANK1
Y[1]
X[1]
W[1]
U[2]
BANK2
Y[2]
X[2]
W[2]
U[3]
BANK3
Y[3]
X[3]
V
W[3]

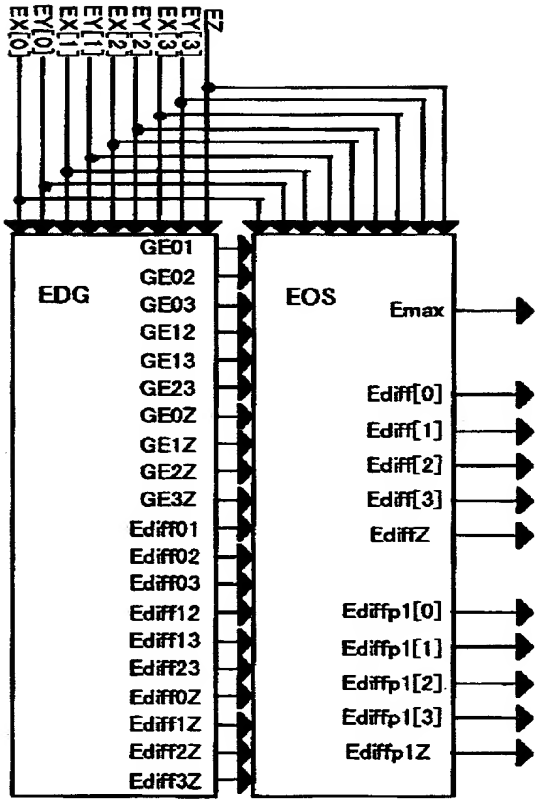
【図５】

図5


EBLK
X[0]
Y[0]
X[1]
Y[1]
MLP0
MLP1
MC[0]
MS[0]
MC[1]
MS[1]
ALN0
ALN1
MCaln[0]
MSaln[0]
MCaln[1]
MSaln[1]
EPP
EX[0]
EY[0]
EX[1]
EY[1]
EX[2]
EY[2]
EX[3]
EY[3]
Ediff[0]
Ediff[1]
Emax
Ediff[2]
Ediff[3]
EdiffZ
S[0]
S[1]
S[2]
S[3]
SZ
SPP
Inv[0]
Inv[1]
Inv[2]
Inv[3]
EZ
MZ
ALNZ
MZaln
Z
X[2]
Y[2]
X[3]
Y[3]
MLP2
MLP3
MC[2]
MS[2]
MC[3]
MS[3]
ALN2
ALN3
MCaln[2]
MSaln[2]
MCaln[3]
MSaln[3]
ADDR
NOR
Maxm
V

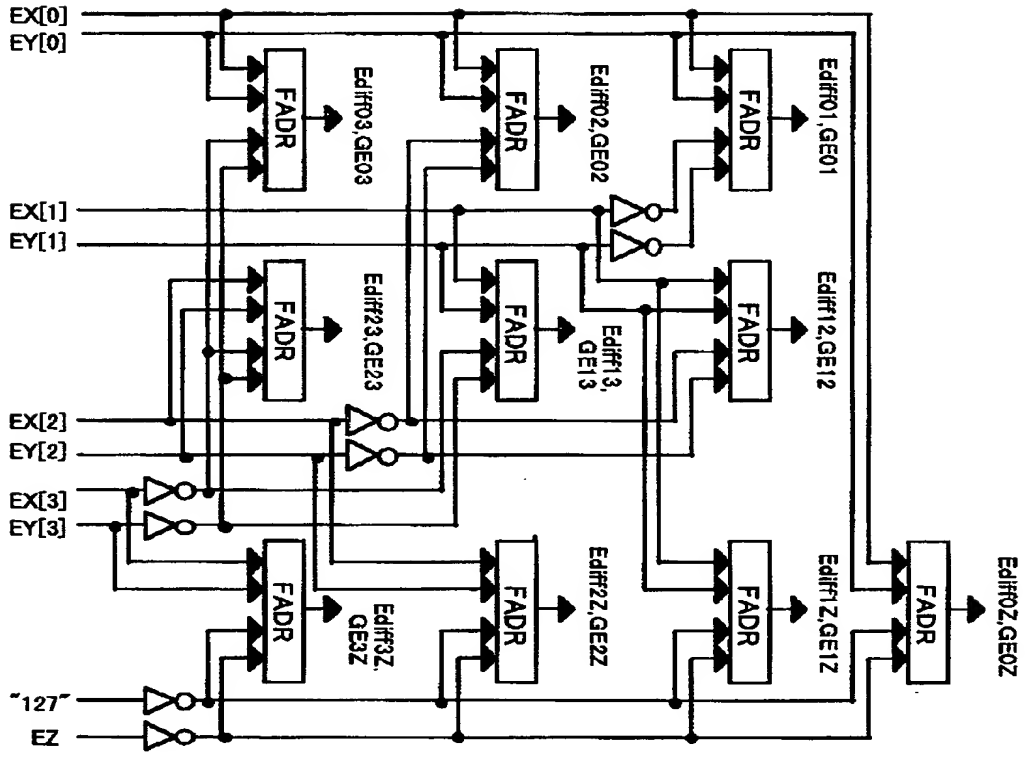
【図６】

図6


EZ
EY[3]
EX[3]
EY[2]
EX[2]
EY[1]
EX[1]
EY[0]
EX[0]
EDG
GE01
GE02
GE03
GE12
GE13
GE23
GE0Z
GE1Z
GE2Z
GE3Z
Ediff01
Ediff02
Ediff03
Ediff12
Ediff13
Ediff23
Ediff0Z
Ediff1Z
Ediff2Z
Ediff3Z
EOS
Emax
Ediff[0]
Ediff[1]
Ediff[2]
Ediff[3]
EdiffZ
Ediffp1[0]
Ediffp1[1]
Ediffp1[2]
Ediffp1[3]
Ediffp1Z

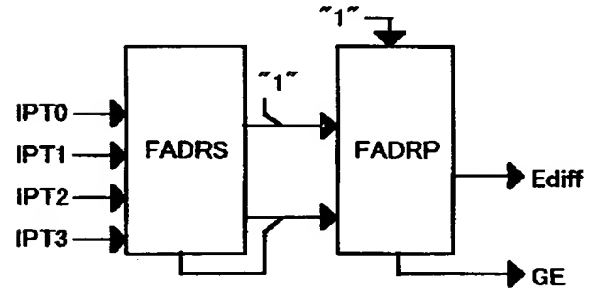
【図７】

図7


EX[0]
EY[0]
EX[1]
EY[1]
EX[2]
EY[2]
EX[3]
EY[3]
"127"
EZ
FADR
Ediff03,GE03
Ediff02,GE02
Ediff01,GE01
Ediff23,GE23
Ediff13,GE13
Ediff12,GE12
Ediff3Z,GE3Z
Ediff2Z,GE2Z
Ediff1Z,GE1Z
Ediff0Z,GE0Z

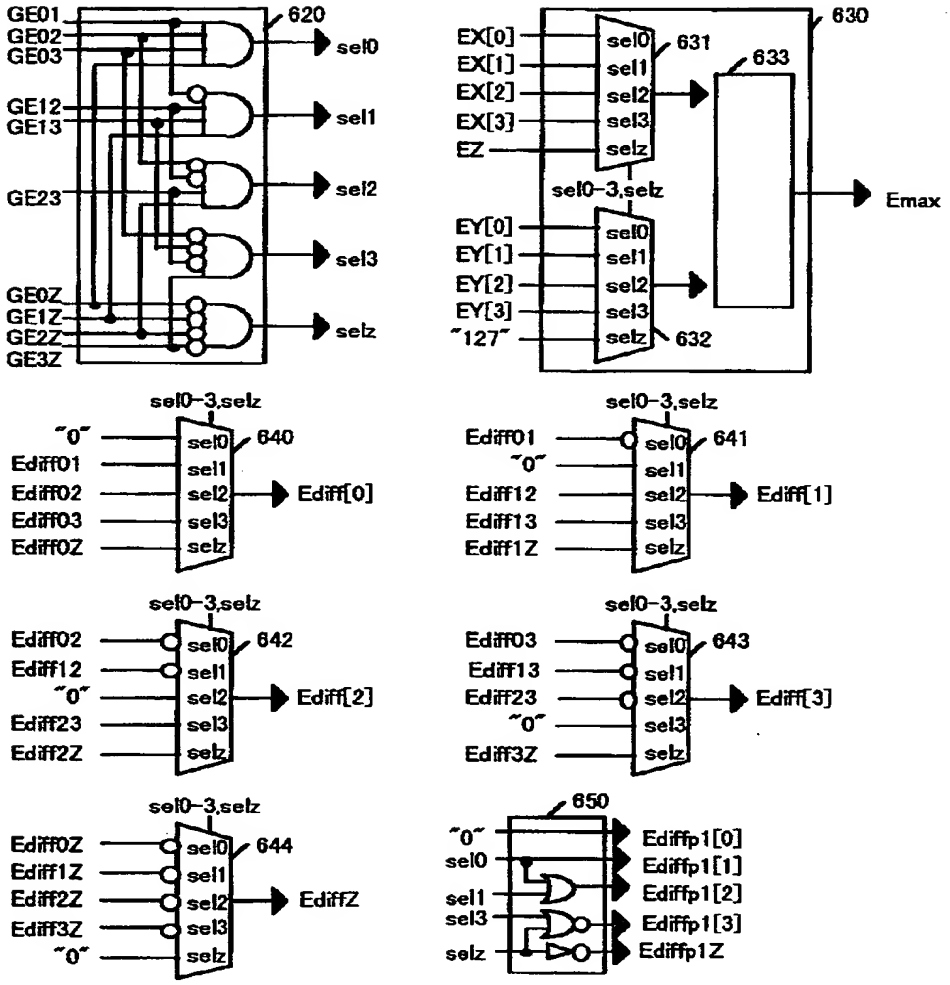
【図８】

図8


"1"
"1"
IPT0
IPT1
IPT2
IPT3
FADRS
FADRP
Ediff
GE

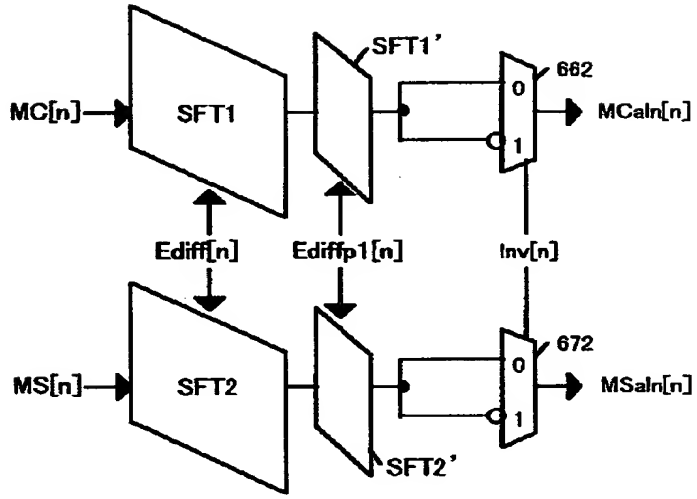
【図９】

図9


GE01
GE02
GE03
GE12
GE13
GE23
GE0Z
GE1Z
GE2Z
GE3Z
620
sel0
sel1
sel2
sel3
selz
EX[0]
EX[1]
EX[2]
EX[3]
EZ
sel0-3,selz
631
633
630
Emax
EY[0]
EY[1]
EY[2]
EY[3]
"127"
632
"0"
Ediff01
Ediff02
Ediff03
Ediff0Z
640
Ediff[0]
Ediff12
Ediff13
Ediff1Z
641
Ediff[1]
Ediff23
Ediff2Z
642
Ediff[2]
Ediff3Z
643
Ediff[3]
Ediff1Z
644
EdiffZ
650
Ediffp1[0]
Ediffp1[1]
Ediffp1[2]
Ediffp1[3]
Ediffp1Z

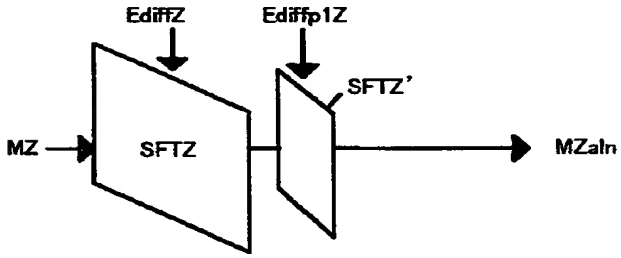
【図１０】

図10



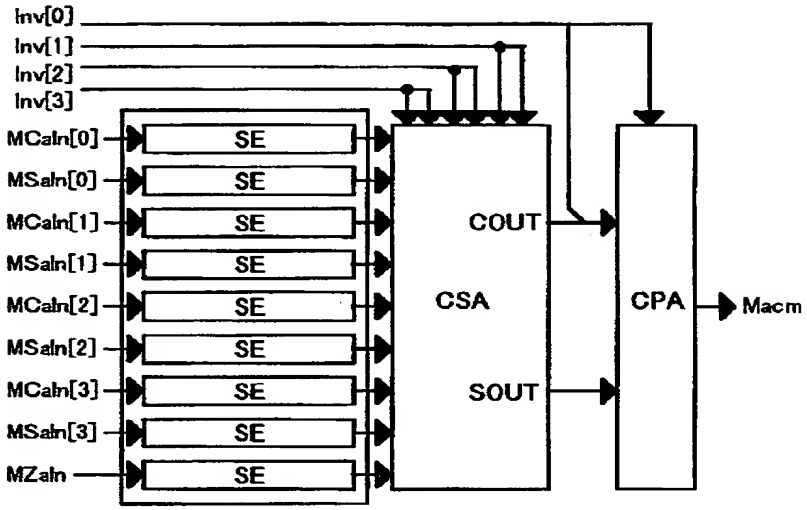
【図１１】

図11



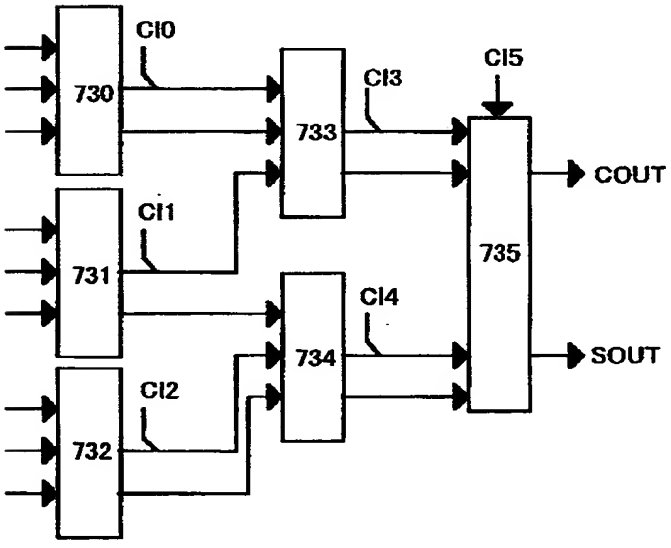
【図１２】

図12



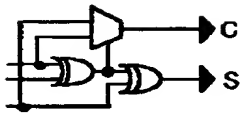
【図１３】

図13



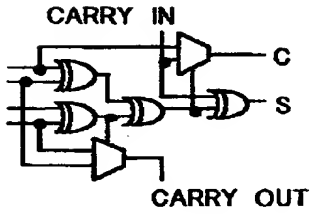
【図１４】

図14



【図１５】

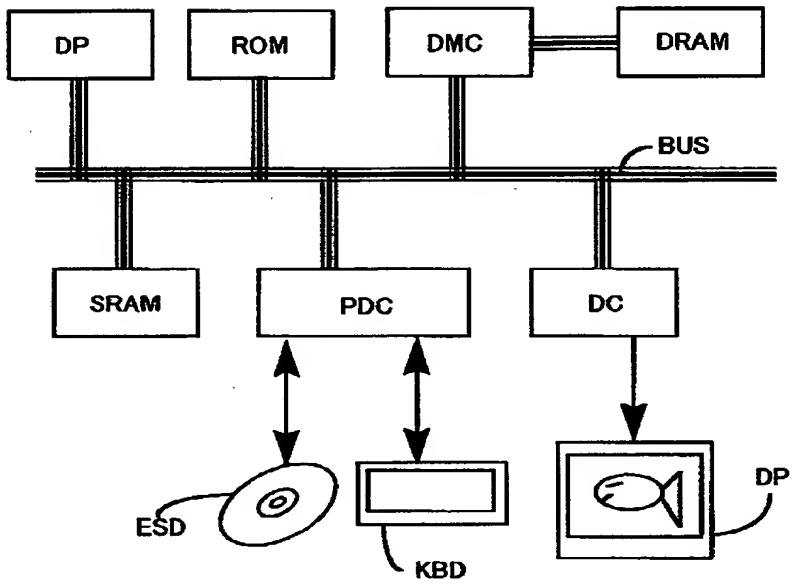
図15



出証特２０００－３０８１６６８

【図１６】

図16



【書類名】要約書

【要約】　浮動小数点内積演算器をＳＩＭＤ化することにより演算並列度を向上させ、演算処理能力高めたデータプロセッサの提供を可能とする。

【課題】　浮動小数点４元ベクトル内積演算器の効率性を維持したまま１命令当りの演算並列度を飛躍的に高める演算方式を実現すること。

【解決手段】　浮動小数点４元ベクトル内積演算器を幅広い演算系の利用が可能になっても、最小幅（単精度なら３２ビット）で定義し、内積演算器をＳＩＭＤ化する。

【効果】　内積演算器とＳＩＭＤ化との双方によるによる並列度向上の相乗効果で演算並列度を飛躍的に高めた。４元ベクトルの内積とスカラとの和を求める浮動小数点演算器を４並列ＳＩＭＤ化すると１サイクル当り３２ＦＬＯＰＳとなる。

【選択図】図２

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所